CLAIMS

We claim:

1. A method for detecting access to structures in a common memory space shared by a plurality of virtual processors included within a virtual machine comprising the following steps:

establishing a trace on at least one traced portion of the common memory space;

associating the trace with a first set of at least one of the virtual processors; and

sensing the occurrence of a trace event arising from operation of an acting one of the virtual processors and thereupon issuing a trace notification to each virtual processor in the first set.

2. A method as in claim 1, in which the first set of virtual processors includes each virtual processor that has any portion of a primary structure stored in the traced portion of the common memory space.

3. A method as in claim 2, in which:

with respect to the traced portion of the common memory space, the first set of virtual processors includes fewer than the total number of the virtual processors, those virtual processors not included in the first set forming a second set; and

issuing the trace notification to only the virtual processors in the first set.

4. A method as defined in claim 2, further including the step of deriving a secondary structure from and corresponding to the primary structure.

5. A method as in claim 4, in which:

each primary structure is accessible to the virtual machine;

each secondary structure is transparent to the virtual machine but is accessible to and operable on at least one hardware processor; and

the trace event is any attempt by any of the virtual processors to access any primary structure stored in any traced portion of the common memory space further comprising the following step:

upon receipt of the trace notification with respect to access to any primary structure, rederiving the corresponding secondary structure, whereby coherence between the primary and secondary structures is maintained.

6. A method as in claim 5, in which the primary structure is original VM code and the secondary structure is translated code derived from the original VM code through binary translation.

7. A method as in claim 5, in which the primary structure is a primary descriptor table and the secondary structure is a shadow descriptor table.

8. A method as in claim 1, in which the acting virtual processor is also in the first set of virtual processors.

9. A method as in claim 1, in which the trace is a write trace and the trace event is any attempt by any of the virtual processors to write to the traced portion of the common memory space, the virtual processor attempting to write thereby becoming the acting virtual processor.

10. A method as in claim 9, further including the following steps:

issuing a pre-write notification to the virtual processors in the first set;

for each of the virtual processors in the first set, acknowledging receipt of the pre-write notification at a virtual instruction boundary and suspending execution of the virtual processors in the first set;

after acknowledgment of receipt of the pre-write notification by all of the virtual processors in the first set allowing the acting virtual processor to complete writing to the traced portion of the common memory space;

after completion of writing by the acting virtual processor, resuming execution of the virtual processors in the first set.

11. A method as in claim 9, further including the following steps:

allowing the acting virtual processor to complete writing to the traced portion of the common memory space; and

issuing a post-write notification to the virtual processors in the first set.

12. A method as in claim 11, further including the following steps:

after completion of the write by the acting virtual processor, idling the acting virtual processor until receipt of acknowledgement from the virtual processors in the first set of their receipt of the post-write notification; and

immediately resuming execution of the acting virtual processor after the acknowledgement of the post-write notification.

13. A method as in claim 1, further comprising the following steps:

running each virtual processor exclusively on a respective hardware processor;

issuing the trace notification to each virtual processor by issuing a conventional fault directly to the hardware processor on which the acting virtual processor is running and thereupon sending a physical interprocessor interrupt (IPI) signal from the hardware processor on which the acting virtual processor is running to the respective hardware processor(s) on which the virtual processor(s) in the first set are running.

14. A method for detecting access to structures in a common memory space shared by a plurality of virtual processors included within a virtual machine comprising the following steps:

establishing a trace on at least one traced portion of the common memory space;

associating the trace with a first set of at least one of the virtual processors; and

sensing the occurrence of a trace event arising from operation of an acting one of the virtual processors and thereupon issuing a trace notification to each virtual processor in the first set;

in which:

the first set of virtual processors includes each virtual processor that has any portion of a primary structure stored in the traced portion of the common memory space;

with respect to the traced portion of the common memory space, the first set of virtual processors includes fewer than the total number of the virtual processors, those virtual processors not included in the first set forming a second set;

issuing the trace notification to only the virtual processors in the first set; and

the trace is a write trace and the trace event is any attempt by any of the virtual processors to write to the traced portion of the common memory space, the virtual processor attempting to write thereby becoming the acting virtual processor.

15. A computer system comprising:

at least one virtual machine (VM) including a plurality of virtual processors that share a common memory space;

a hardware platform including a physical memory and at least one physical processor that executes instructions originating in the virtual machine;

interface means:

for providing for execution of VM-issued instructions and for transferring data between the VM and the physical memory;

for establishing a trace on at least one traced portion of the common memory space;

for associating the trace with a first set of at least one of the virtual processors; and

for sensing the occurrence of a trace event arising from operation of an acting one of the virtual processors and thereupon for issuing a trace notification to each virtual processor in the first set.

16. A system as in claim 15, in which the first set of virtual processors includes each virtual processor that has any portion of a primary structure stored in the traced portion of the common memory space.

17. A system as in claim 16, in which:

with respect to the traced portion of the common memory space, the first set of virtual processors includes fewer than the total number of the virtual processors, those virtual processors not included in the first set forming a second set; and

the interface means is further provided for issuing the trace notification to only the virtual processors in the first set.

18. A system as in claim 15, in which the acting virtual processor is also in the first set of virtual processors.

19. A system as in claim 15, in which the trace is a write trace and the trace event is any attempt by any of the virtual processors to write to the traced portion of the common memory space, the virtual processor attempting to write thereby becoming the acting virtual processor.

20. A system as in claim 19, in which:

the interface means is further provided:

for issuing a pre-write notification to the virtual processors in the first set;

for each of the virtual processors in the first set, for acknowledging receipt of the pre-write notification at a virtual instruction boundary and suspending execution of the virtual processors in the first set;

after acknowledgment of receipt of the pre-write notification by all of the virtual processors in the first set, for allowing the acting virtual processor to complete writing to the traced portion of the common memory space;

after completion of writing by the acting virtual processor, for resuming execution of the virtual processors in the first set.

21. A system as in claim 15, in which the interface means is a virtual machine monitor (VMM).

22. A system as in claim 21, further comprising a separate sub-monitoring means within the interface means for each virtual processor, each sub-monitoring means being provided:

for establishing each trace on any portion of the common memory space used for storage of any primary structure by its respective virtual processor; and

for sensing the occurrence of the trace event arising from operation of its respective virtual processor and for issuing the corresponding trace notification.